Exercise walk-through

4.1.1 Registers, interrupts, ports and performance

A-Level Computer Science

You must be able to

- explain the Von Neumann model for a computer system and the stored program concept
- state the role of each special purpose register, and the difference between general purpose and special purpose registers
- describe the roles of the Arithmetic and Logic Unit (ALU), the Control Unit (CU), the system clock and the Immediate Access Store (IAS)
- describe how the address bus, data bus and control bus transfer data, and work out the effect of a bus width
- describe the stages of the fetch-execute cycle in register transfer notation
- explain the causes and uses of interrupts, when they are detected, and how an Interrupt Service Routine (ISR) handles them
- explain how processor type, the number of cores, bus width, clock speed and cache memory affect performance

1 The method

Method — The stored program concept and the registers

In the **Von Neumann** model, program instructions and data are stored together in the same memory, the **Immediate Access Store** (main memory). The processor fetches the instructions from memory one at a time and executes them in order: this is the **stored program concept**.

register	purpose
Program Counter (PC)	holds the address of the next instruction to be fetched
Memory Address Register (MAR)	holds the address of the memory location to be read or written
Memory Data Register (MDR)	holds the data or instruction just read from memory, or about to be written to it
Current Instruction Register (CIR)	holds the instruction being decoded and executed
Accumulator (ACC)	holds the value being worked on and the result of each ALU operation
Index Register (IX)	holds a value that is added to an address, to step through a list of memory locations
Status Register (SR)	holds flags, such as carry, negative, zero and overflow, set by the result of an operation

Method — The stored program concept and the registers

- A **special purpose register** has one fixed job in the fetch-execute cycle. A **general purpose register**, such as the ACC in many processors, holds whatever values a program is working on.
- The **system clock** sends timing signals that keep the components working in step. The **Control Unit** decodes each instruction and sends control signals, and the **ALU** carries out arithmetic and logic operations.

Method — Register transfer notation

`[PC]` means the contents of the PC, and ← means "is copied into". `[[MAR]]` means the contents of the memory location whose address is held in the MAR.

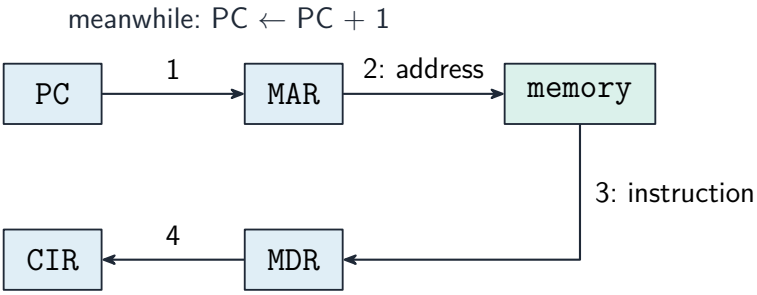
step	register transfer notation	in words
1	`MAR ← [PC]`	the address of the next instruction is copied from the PC to the MAR
2	`PC ← [PC] + 1`	the PC is increased by 1, so it points at the next instruction
3	`MDR ← [[MAR]]`	the instruction at that address is copied into the MDR
4	`CIR ← [MDR]`	the instruction is copied into the CIR, ready to be decoded

Method — Register transfer notation

- Step 2 may also come after step 3: the PC only has to be increased before the next fetch.
- After the fetch, the Control Unit **decodes** the instruction in the CIR, and then it is **executed**.

Method — Register transfer notation

[PC] means the contents of the PC, and ← means "is copied into".



Method — Buses and bus width

bus	carries	direction
address bus	memory addresses, from the processor to memory	one-way
data bus	data and instructions, between the processor, memory and devices	two-way
control bus	control signals, such as memory read, memory write, clock and interrupt signals	two-way

Method — Buses and bus width

- An address bus with n lines can carry 2^n different addresses. A 16-bit address bus can address $2^{16} = 65\,536$ locations, and a 32-bit address bus can address 2^{32} locations, which is 4 GiB of bytes.
- A wider **data bus** moves more bits in each transfer, so more data moves per second.

Method — Interrupts

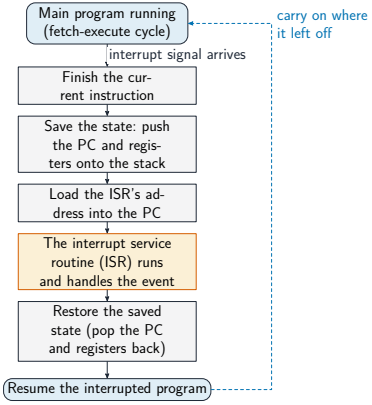
An **interrupt** is a signal from hardware or software that tells the processor something needs attention.

- **Hardware** causes: a key is pressed, a printer runs out of paper, a timer signal.
- **Software** causes: division by zero, an attempt to use a memory location that is not allowed, a request for an operating system service.

1. The processor checks for interrupts at the **end of each fetch-execute cycle**, before the next instruction is fetched.
2. If the interrupt has a higher priority than the current task, the contents of the PC and the other registers are saved on the **stack** 栈.
3. The address of the **Interrupt Service Routine** 中断服务程序 (ISR) is loaded into the PC, and the ISR runs and deals with the event.
4. The saved contents are restored, and the interrupted program carries on where it stopped.

Method — Interrupts

An interrupt is a signal from hardware or software that tells the processor something needs attention



Method — Performance and ports

factor	**why it affects performance**
processor type	a newer design does more work in each clock cycle
number of cores	each core fetches and executes instructions, so several tasks run at the same time
bus width	a wider data bus moves more bits per transfer; a wider address bus allows more memory
clock speed	more clock cycles per second means more instructions per second
cache memory	fast memory next to the processor means fewer slow fetches from main memory

Method — Performance and ports

- More cores only help when the software can divide its work between them.

port	carries	typical use
Universal Serial Bus (USB)	data both ways, and power; plug and play	keyboard, mouse, flash drive, external hard disk, printer
High Definition Multimedia Interface (HDMI)	high definition digital video and sound	monitor, television, projector
Video Graphics Array (VGA)	analogue video only, no sound	older monitors and projectors

2 Practice

Practice 2.1 ▪ 2 marks

Explain what is meant by the **stored program concept**.

Solution 2.1

Method: The stored program concept and the registers — p.4

Worked steps

- program instructions and data are stored together in the same memory B1
- the instructions are fetched from memory and executed one at a time, in order B1

Practice 2.2 ▪ 2 marks

Describe the step `MDR ← [[MAR]]`.

Solution 2.2

Worked steps

- the contents of the memory location whose address is held in the MAR B1
- are copied into the MDR B1

Practice 2.3 ▪ 2 marks

State the purpose of the **Status Register** and of the **Index Register**.

Solution 2.3

Method: The stored program concept and the registers — p.4

Worked steps

- **Status Register:** holds flags, such as carry, zero, negative and overflow, set by the result of an operation B1
- **Index Register:** holds a value that is added to an address, to step through memory locations B1

Practice 2.4 ▪ 2 marks

A processor has a 24-bit address bus. Calculate how many different memory locations it can address.

Solution 2.4

Method: Buses and bus width — p.9

Worked steps

- n lines give 2^n addresses **M1**
- $2^{24} = \mathbf{16\,777\,216}$ locations **A1**

Practice 2.5 ▪ 3 marks

Identify the most suitable port for each device.

Practice 2.5 (a) ▪ 1 mark

a keyboard

Solution 2.5 (a)

> Method: Performance and ports — p.13

Worked steps

USB B1

Practice 2.5 (b) ▪ 1 mark

a television that must receive high definition video and sound

Solution 2.5 (b)

Worked steps

HDMI B1

Practice 2.5 (c) ▪ 1 mark

an old projector with an analogue video input

Solution 2.5 (c)

Worked steps

VGA B1

Practice 2.6 ▪ 3 marks

Give **two** hardware causes and **one** software cause of an interrupt.

Solution 2.6

Method: Interrupts — p.11

Worked steps

- hardware, any two of: a key is pressed; a printer runs out of paper; the mouse is moved; a timer signal B1 B1
- software, one of: division by zero; an attempt to use a memory location that is not allowed; a request for an operating system service B1

3 Exam-style

Exam-style 3.1 ▪ 3 marks

Complete the register transfer notation for the fetch stage of the fetch-execute cycle.

step	register transfer notation
1	MAR ← [PC]
2	
3	
4	

Solution 3.1

Method: Register transfer notation — p.6

Worked steps

- step 2: `PC ← [PC] + 1` B1
- step 3: `MDR ← [[MAR]]` B1
- step 4: `CIR ← [MDR]` B1

Exam-style 3.2 ▪ 8 marks

A student is typing a report when the printer, which is printing another document, runs out of paper.

Exam-style 3.2 (a) ▪ 1 mark

State whether this causes a hardware interrupt or a software interrupt, and give a reason.

Solution 3.2 (a)

Method: Interrupts — p.11

Worked steps

a **hardware** interrupt, because the signal comes from a device, the printer B1

Exam-style 3.2 (b) ▪ 5 marks

Explain how the processor detects and handles this interrupt.

Solution 3.2 (b)

Worked steps

any **five** of:

- the printer sends an interrupt signal to the processor B1
- the processor checks for interrupts at the end of each fetch-execute cycle B1
- it compares the priority of the interrupt with the priority of the current task B1
- the contents of the PC and the other registers are saved on the stack B1
- the address of the Interrupt Service Routine is loaded into the PC B1
- the ISR runs, for example to show a message asking for more paper B1
- the saved contents are restored, and the report program carries on B1

Exam-style 3.2 (c) ▪ 2 marks

Explain why the contents of the registers must be saved.

Solution 3.2 (c)

Worked steps

the registers hold the state of the interrupted program, including the address of its next instruction in the PC B1

- restoring them lets the program carry on exactly where it stopped, with nothing lost B1

Exam-style 3.3 ▪ 5 marks

Two computers are for sale.

	computer A	**computer B**
number of cores	4	8
clock speed	3.6 GHz	3.0 GHz
cache memory	8 MB	16 MB

A video-editing company wants the computer that will edit video faster. **Explain** how **each** of the three factors affects performance, and state which computer the company should choose.

Solution 3.3

Method: Performance and ports — p.13

Worked steps

- **cores:** more cores execute more instructions at the same time, and video editing software can divide its work between them, so B gains B1 B1
- **clock speed:** A's faster clock completes more fetch-execute cycles per second on each core B1
- **cache:** B's larger cache holds more of the instructions and data in use, so fewer slow fetches from main memory are needed B1
- choose **computer B**: for video editing, eight cores and more cache outweigh A's faster clock B1

Exam-style 3.4 (a) ▪ 1 mark

State the number of different memory locations that can be addressed with a 20-bit address bus.

Solution 3.4 (a)

Method: Buses and bus width — p.9

Worked steps

$2^{20} = \mathbf{1\,048\,576}$ B1

Exam-style 3.4 (b) ▪ 2 marks

Explain the effect of doubling the width of the data bus.

Solution 3.4 (b)

Worked steps

twice as many bits are carried in each transfer B1

- so more data is moved per second, and larger values can be moved in one step B1

Exam-style 3.4 (c) ▪ 2 marks

Identify **two** signals carried on the control bus.

Solution 3.4 (c)

Worked steps

any two of: memory read; memory write; clock signal; interrupt request; bus request B1 B1

Exam-style 3.5 ▪ 5 marks

A student connects a monitor, a keyboard and an external hard disk to a laptop.

Exam-style 3.5 (a) ▪ 3 marks

Identify a suitable port for each device.

Solution 3.5 (a)

Method: Performance and ports — p.13

Worked steps

monitor: **HDMI** B1; keyboard: **USB** B1; external hard disk: **USB** B1

Exam-style 3.5 (b) ▪ 2 marks

The monitor also has a VGA input. **Explain** why the picture may be worse through the VGA port than through the HDMI port.

Solution 3.5 (b)

Worked steps

VGA carries an **analogue** signal, so the digital picture must be converted, and it can pick up noise and lose detail B1

- HDMI carries a **digital**, high definition signal, and it carries sound as well B1